

Лекция №4. Пример проекта

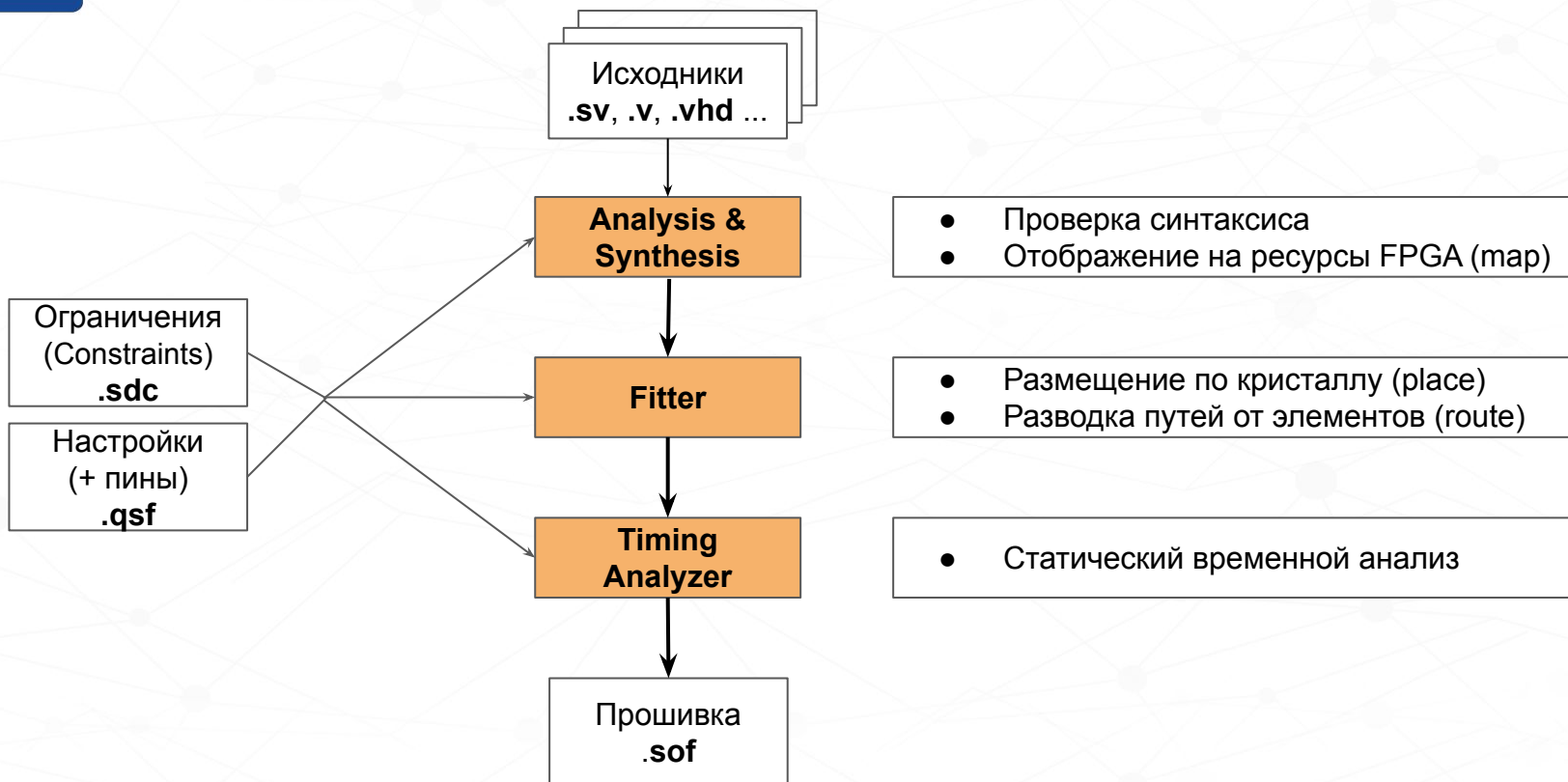
Толкачев Максим

m.tolkachev@metrotek.ru

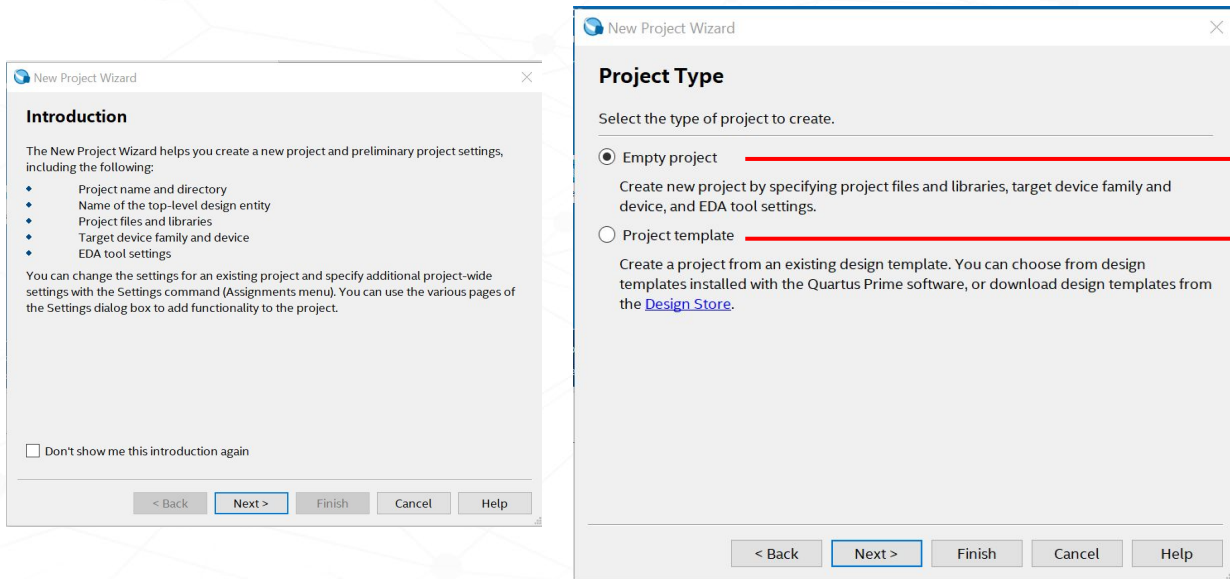
- I. Процесс создания и сборки проекта в Quartus
- II. Временные характеристики схемы
- III. Временные ограничения (Timing Constraints)
- IV. Временной анализатор в Quartus (Timing Analyzer)
- V. Настройки проекта Quartus, назначение пинов
- VI. Собираем проект игры “N-back”
 - Правила игры
 - DE0-Nano
 - Смотрим как игра работает
 - Разбираемся в проекте
 - Собираем проект в Quartus
 - Играем
 - Добавляем отладчик

I. Процесс создания и сборки проекта в Quartus

Этапы сборки проекта в Quartus



Инструкция от Intel на YouTube: [Creating a New Project with Intel® Quartus® Prime Pro Edition Software](#)



Настройка проекта вручную

Скачиваем шаблон для devkit тут:

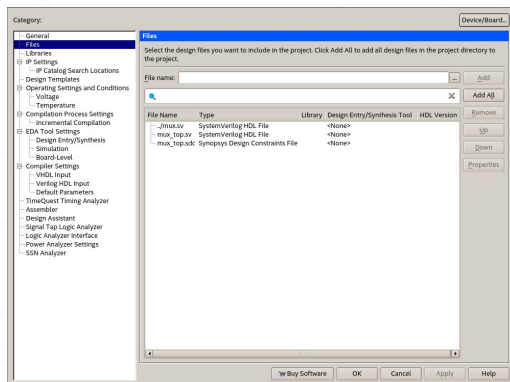
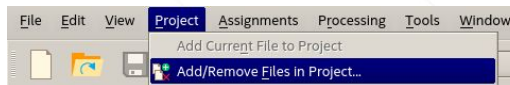
[Design Store for Intel® FPGAs](#)

В шаблоне уже будут:

- топ модуль со всеми сигналами
- назначение сигналов на пины

Этапы создания нового проекта в Quartus

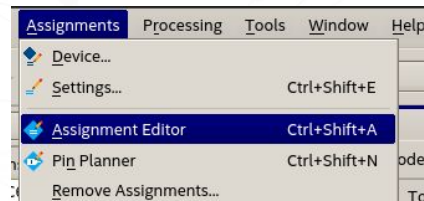
1. Добавить исходники



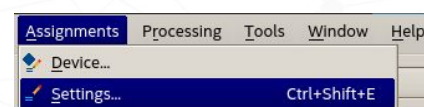
2. Добавить Constraints

- В текстовом редакторе пишем **sdc**
- Добавляем как исходник

3. Добавить пины

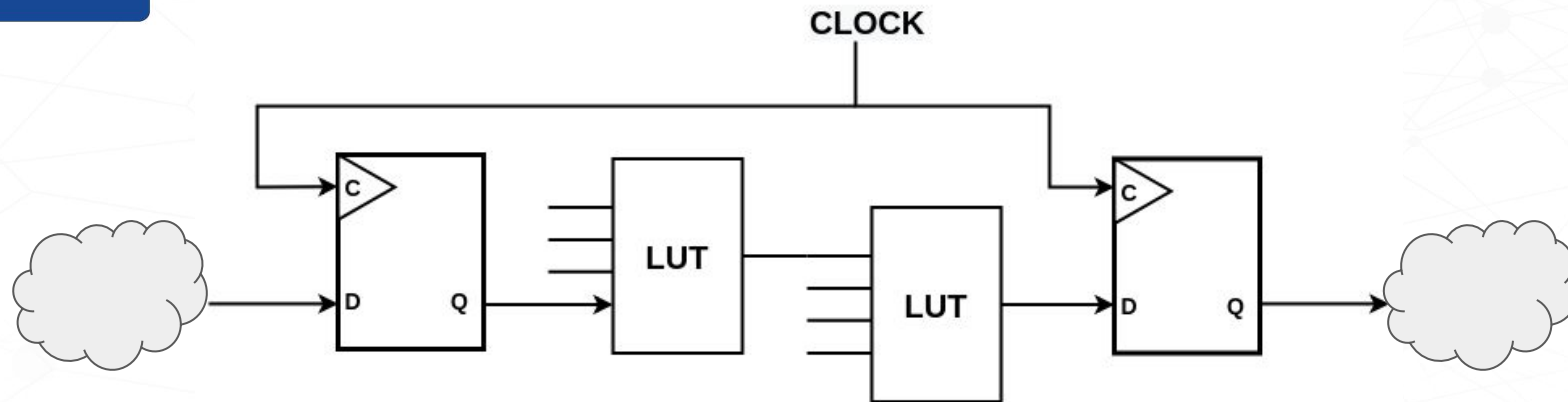


4. Настроить проект

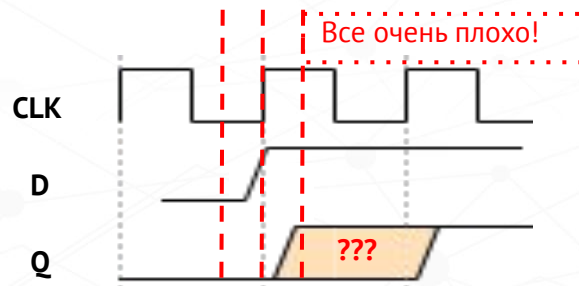
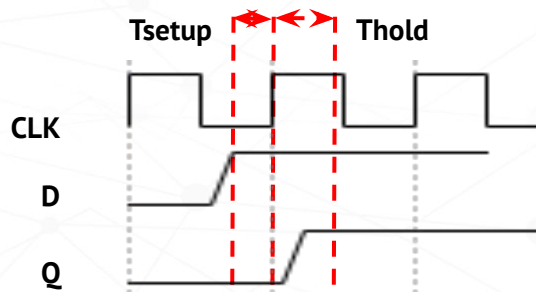


II. Временные характеристики схемы

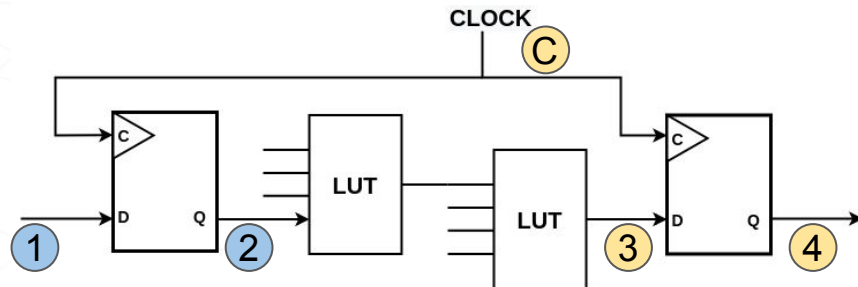
Временные характеристики схемы (внутри)



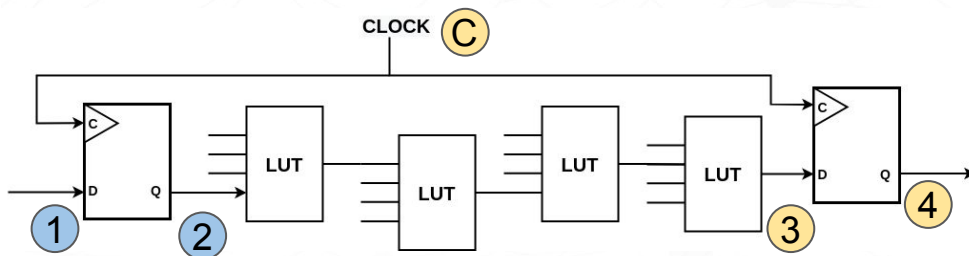
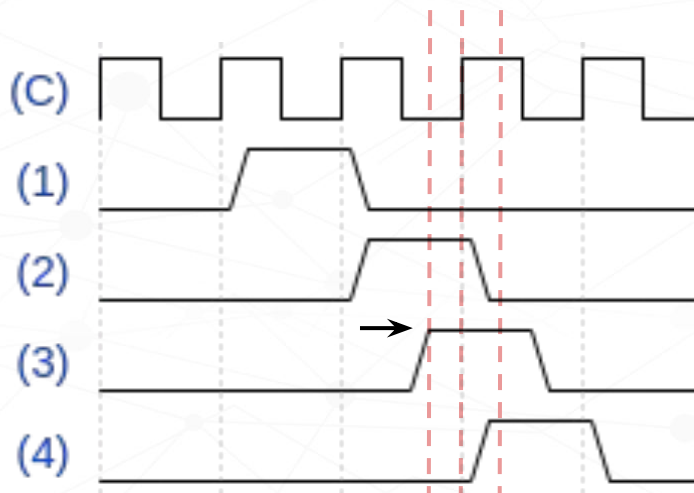
- На все регистры CLOCK приходит почти одновременно
- Для прохождения сигнала через LUT нужно время (и это время зависит от температуры)
- Путь между элементами тоже занимает время (и это время зависит от температуры)
- Сигнал должен прийти на вход следующего триггера чуть раньше следующего фронта CLOCK и держаться после этого немного времени



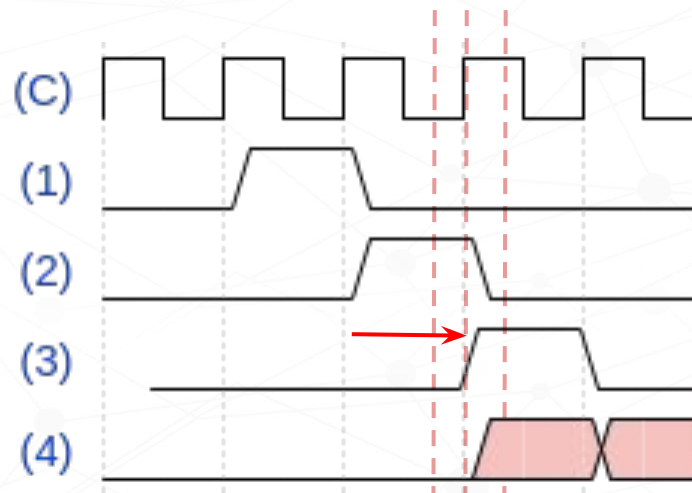
Временные характеристики схемы (внутри)



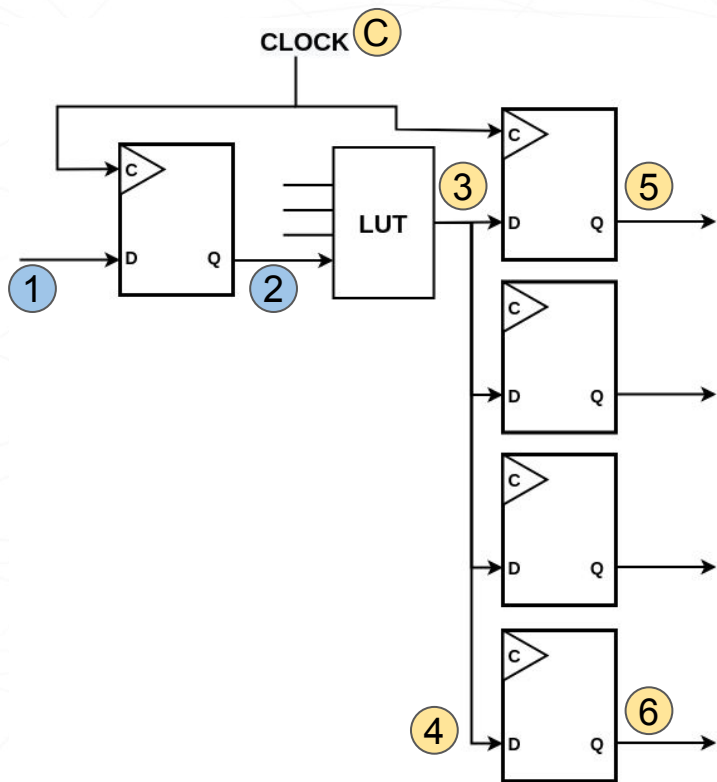
Все хорошо!



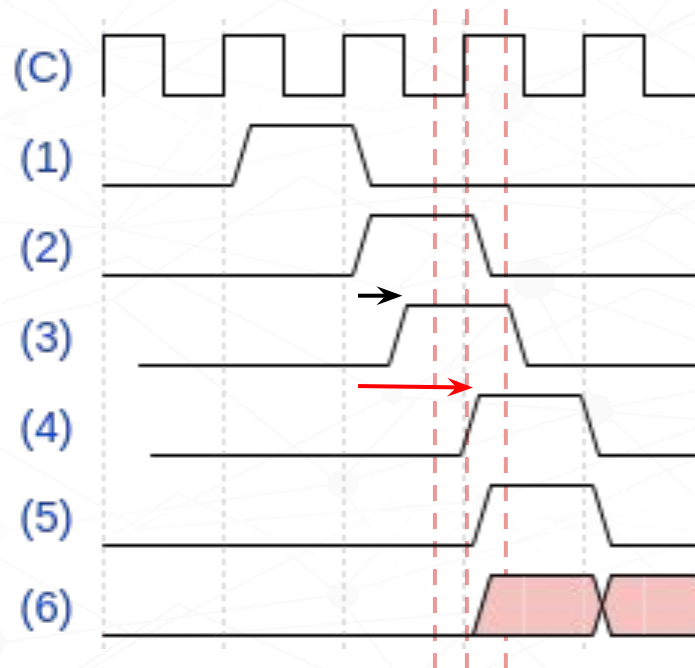
Все плохо! Слишком много логики!

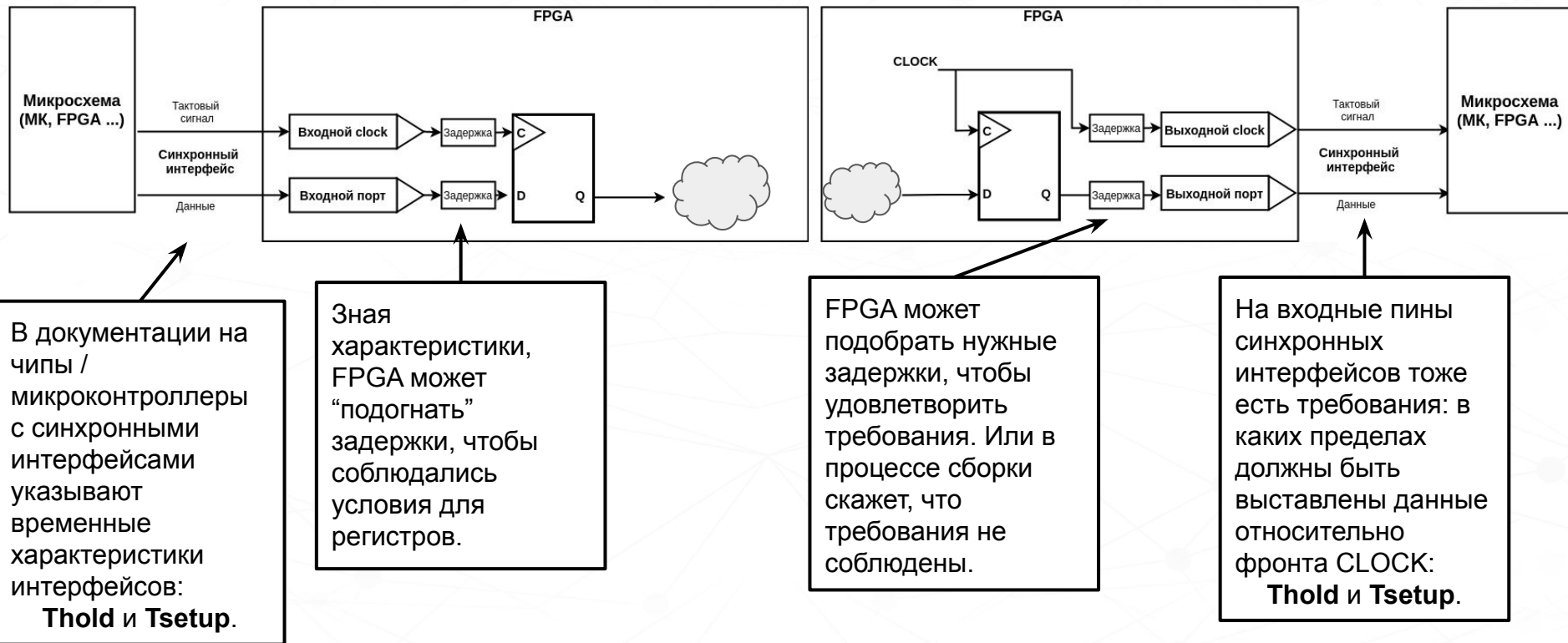


Временные характеристики схемы (внутри)



Все плохо! Слишком длинный путь!





III. Временные ограничения (Timing Constraints)

Ограничения задаются на языке Synopsys Design Constraint в файле с расширением **.sdc**

Минимальный **.sdc** файл:

Формат чисел:

Описание входного
тактового сигнала:

Автоматически
распознавать клок из
PLL:

```
set_time_format -unit ns -decimal_places 3  
  
create_clock -name {clk_100} -period 10.000 -waveform { 0.000 5.000 } [get_ports {clk_i}]  
  
derive_pll_clocks
```

Если проект сложнее, то есть в нем есть, например:

- Несколько тактовых сигналов
- Входной или выходной синхронный интерфейс

То нужно читать дополнительные материалы. Начать стоит с [TimeQuest User Guide By: Ryan Scoville](#)
(будет полезен и пользователям Vivado)

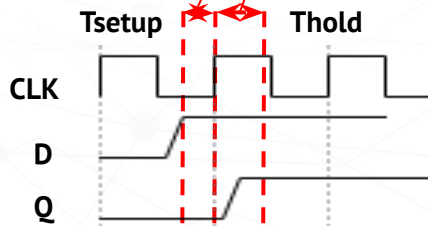
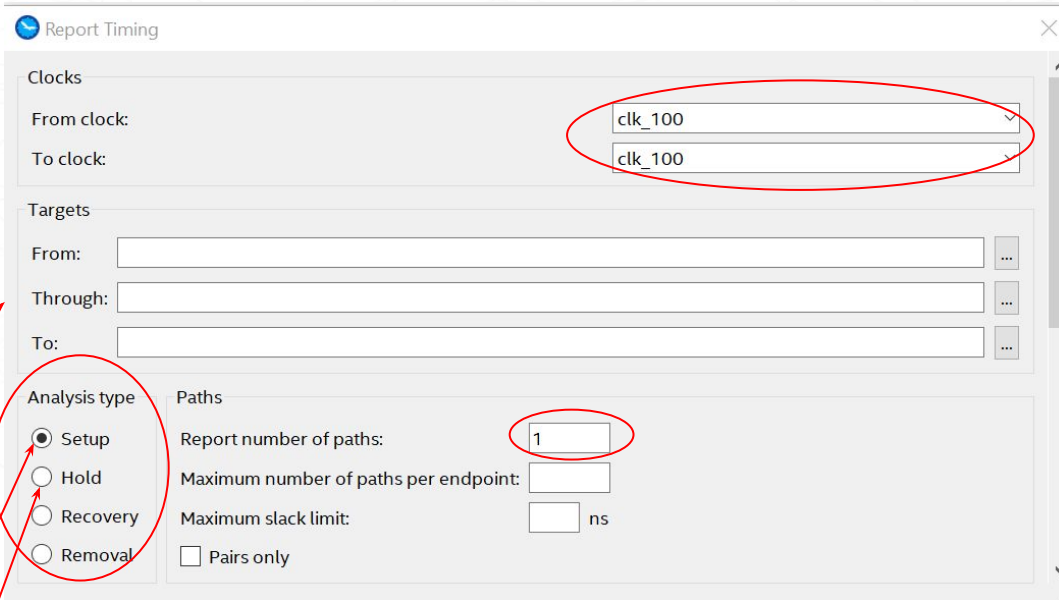
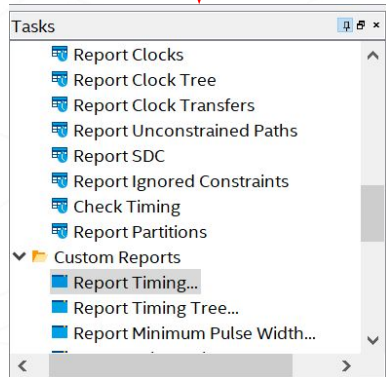
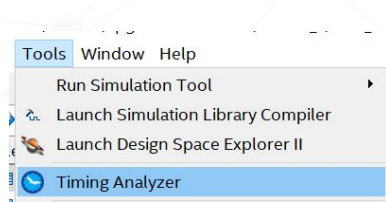
IV. Временной анализатор в Quartus (Timing Analyzer)

Раньше назывался **TimeQuest** теперь называется **Timing Analyzer**.
Основная документация: [Intel Quartus Prime Pro Edition User Guide: Timing Analyzer](#)

Анализ выполняется **статически** -- никакие “сигналы” не подаются, симуляция не проводится.

Анализатор:

- Создает список всех путей между регистрами
- Оценивается разница задержки от тактовых входов до всех регистров
- Для каждого пути рассчитывается **наименьшее** время прохождения сигнала (для лучшего чипа в серии при минимальной рабочей температуре)
- Для каждого пути рассчитывается **наибольшее** время прохождения сигнала (для худшего чипа в серии при максимальной рабочей температуре)



Оценка по **Setup**: сигнал должен успеть дойти до регистра до следующего фронта клона.
Чем больше логики и меньше период клона, тем хуже.

Оценка по **Hold**: сигнал не должен слишком быстро прийти до регистра, чтобы он успел защелкнуть старое значение.
Чем меньше логики тем хуже (период клона никак не влияет).

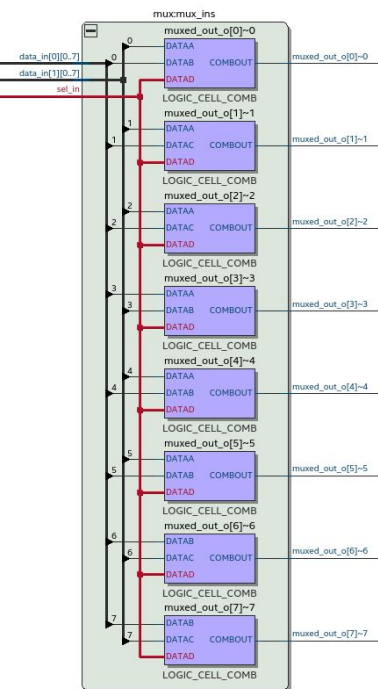
Результаты Timing Analyzer для мультиплексора (из лекции 3):

https://github.com/stcmtk/fpga-webinar-2020/tree/master/lecture_3/mux_demux/mux_example

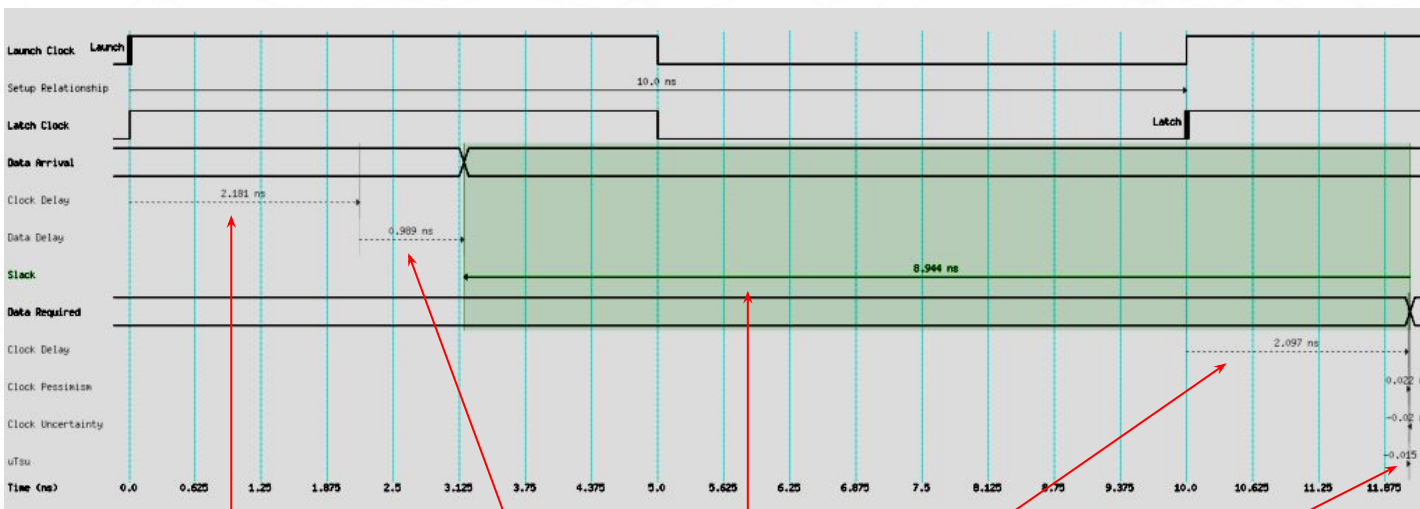
parameter IN_CNT = 2

parameter DATA_W = 8

Clock: 100 МГц



Command Info		Summary of Paths						
	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	8.944	data_in[0][3]	data_out_o[3]~reg0	clk_100	clk_100	10.000	-0.062	0.989
2	8.945	data_in[0][0]	data_out_o[0]~reg0	clk_100	clk_100	10.000	-0.062	0.988
3	8.946	data_in[0][6]	data_out_o[6]~reg0	clk_100	clk_100	10.000	-0.061	0.988



Путь клона от пина до регистра отправления.

Путь от выхода регистра отправления до регистра приема.

Запас

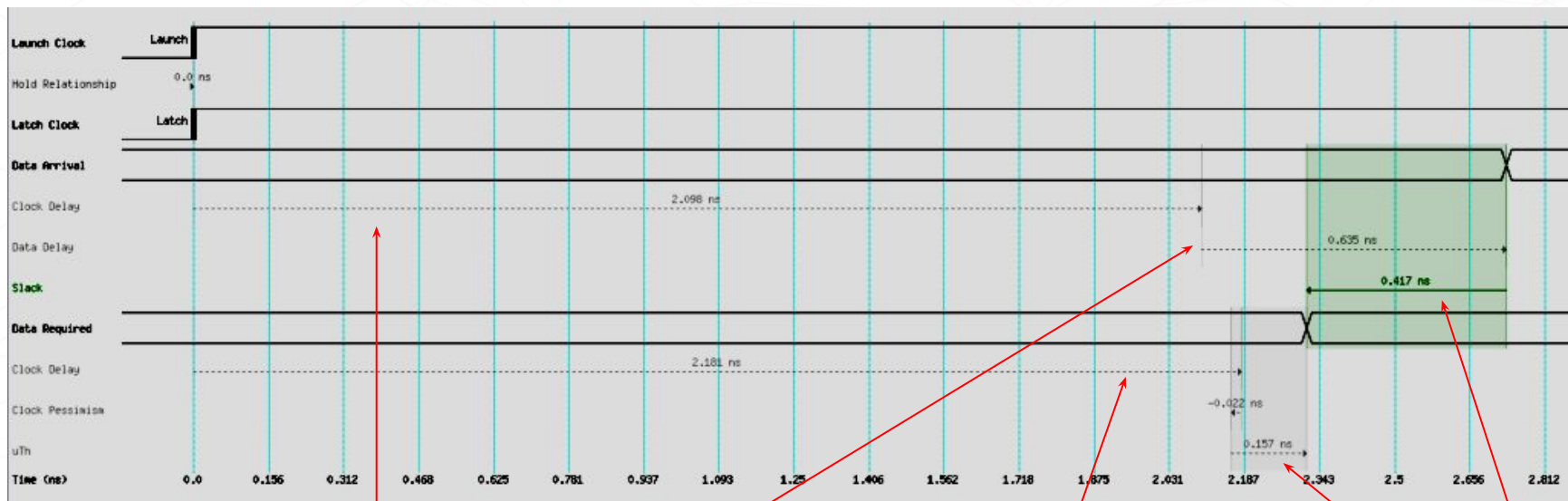
Путь клона от пина до регистра приема.

Tsetup

Полный путь с детализацией задержек:

Path #1: Setup slack is 8.944							
Path Summary		Statistics	Data Path		Waveform	Extra Fitter Information	
Data Arrival Path							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	▼ 2.181	2.181					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_R8	clk_i
3	0.000	0.000	RR	IC	1	IOIBUF_X27_Y0_N22	clk_i~input i
4	0.514	0.514	RR	CELL	1	IOIBUF_X27_Y0_N22	clk_i~input o
5	0.668	0.154	RR	IC	1	CLKCTRL_G18	clk_i~inputclkctrl inclk[0]
6	0.668	0.000	RR	CELL	25	CLKCTRL_G18	clk_i~inputclkctrl outclk
7	1.662	0.994	RR	IC	1	FF_X39_Y12_N11	data_in[0][3] clk
8	2.181	0.519	RR	CELL	1	FF_X39_Y12_N11	data_in[0][3]
3	▼ 3.170	0.989					data path
1	2.380	0.199		uTco	1	FF_X39_Y12_N11	data_in[0][3]
2	2.380	0.000	FF	CELL	1	FF_X39_Y12_N11	data_in[0][3] q
3	2.712	0.332	FF	IC	1	LCCOMB_X39_Y12_N4	mux_ins muxed_out_o[3]~3 dataa
4	3.079	0.367	FF	CELL	1	LCCOMB_X39_Y12_N4	mux_ins muxed_out_o[3]~3 combout
5	3.079	0.000	FF	IC	1	FF_X39_Y12_N5	data_out_o[3]~reg0 d
6	3.170	0.091	FF	CELL	1	FF_X39_Y12_N5	data_out_o[3]~reg0

Command Info		Summary of Paths						
	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	0.417	sel_in	data_out_o[7]~reg0	clk_100	clk_100	0.000	0.061	0.635
2	0.419	sel_in	data_out_o[6]~reg0	clk_100	clk_100	0.000	0.061	0.637
3	0.419	sel_in	data_out_o[5]~reg0	clk_100	clk_100	0.000	0.061	0.637



Путь клок от пина до регистра отправления.

Путь от выхода регистра отправления до регистра приема.

Путь клок от пина до регистра приема.

Thold

Запас

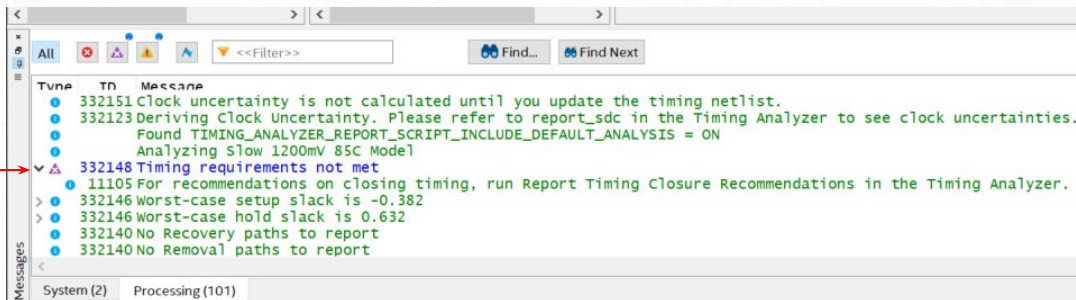
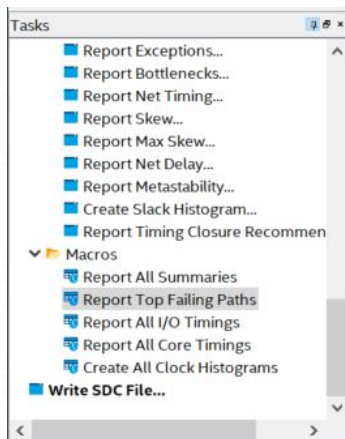
Полный путь с детализацией задержек:

Path #1: Hold slack is 0.417							
Path Summary		Statistics	Data Path		Waveform	Extra Fitter Information	
Data Arrival Path							
	Total	Incr	RF	Type	Fanout	Location	Element
2	▼ 2.098	2.098					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_R8	clk_i
3	0.000	0.000	RR	IC	1	IOIBUF_X27_Y0_N22	clk_i~input i
4	0.514	0.514	RR	CELL	1	IOIBUF_X27_Y0_N22	clk_i~input o
5	0.662	0.148	RR	IC	1	CLKCTRL_G18	clk_i~inputclkctrl inclk[0]
6	0.662	0.000	RR	CELL	25	CLKCTRL_G18	clk_i~inputclkctrl outclk
7	1.617	0.955	RR	IC	1	FF_X40_Y12_N19	sel_in clk
8	2.098	0.481	RR	CELL	1	FF_X40_Y12_N19	sel_in
3	▼ 2.733	0.635					data path
1	2.297	0.199		uTco	1	FF_X40_Y12_N19	sel_in
2	2.297	0.000	RR	CELL	8	FF_X40_Y12_N19	sel_in q
3	2.552	0.255	RR	IC	1	LCCOMB_X40_Y12_N24	mux_ins muxed_out_o[7]~7 datad
4	2.667	0.115	RF	CELL	1	LCCOMB_X40_Y12_N24	mux_ins muxed_out_o[7]~7 combout
5	2.667	0.000	FF	IC	1	FF_X40_Y12_N25	data_out_o[7]~reg0 d
6	2.733	0.066	FF	CELL	1	FF_X40_Y12_N25	data_out_o[7]~reg0

Значит со
схемой
проблемы.
Нужно
чинить.

parameter IN_CNT = 16
parameter DATA_W = 512

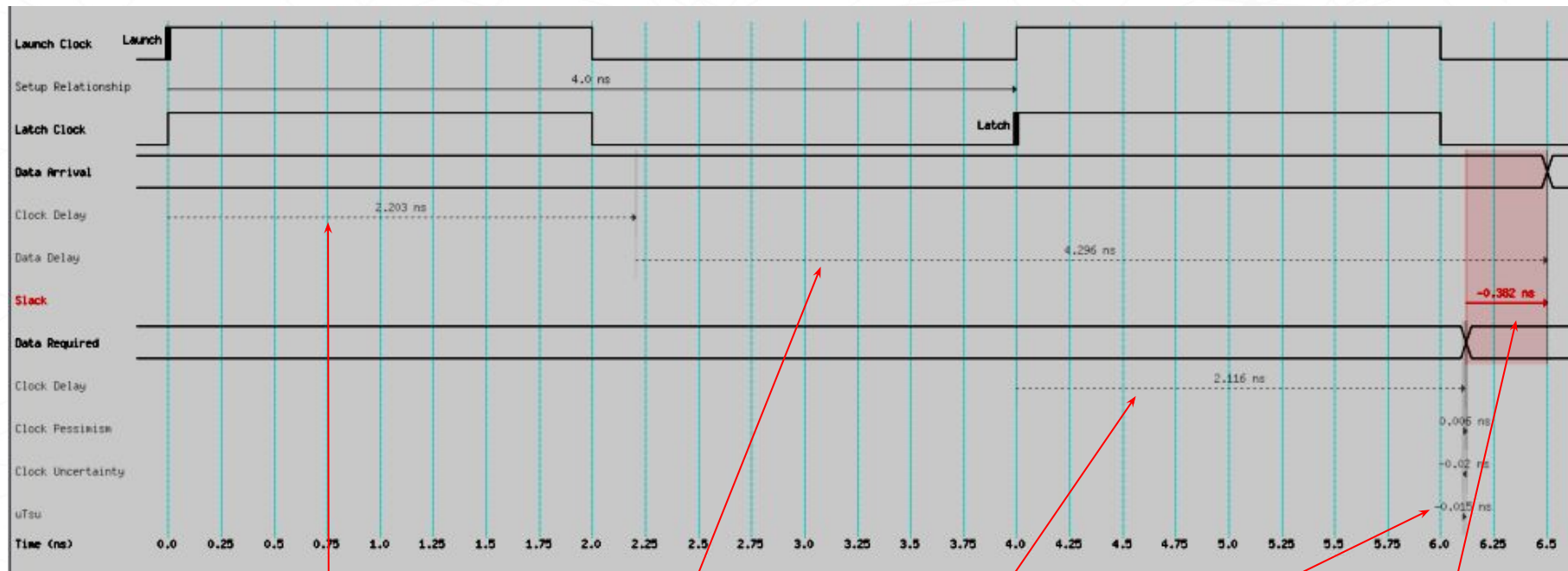
Clock: 250 МГц



	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-0.382	sel_in[2]	data_out_o[313]~reg0	clk_250	clk_250	4.000	-0.081	4.296
2	-0.236	sel_in[0]	data_out_o[266]~reg0	clk_250	clk_250	4.000	-0.070	4.161
3	-0.233	sel_in[3]	data_out_o[61]~reg0	clk_250	clk_250	4.000	-0.082	4.146
4	-0.199	sel_in[3]	data_out_o[481]~reg0	clk_250	clk_250	4.000	-0.067	4.127
5	-0.186	sel_in[2]	data_out_o[315]~reg0	clk_250	clk_250	4.000	-0.090	4.091
6	-0.170	sel_in[2]	data_out_o[487]~reg0	clk_250	clk_250	4.000	-0.083	4.082
7	-0.144	sel_in[0]	data_out_o[42]~reg0	clk_250	clk_250	4.000	-0.082	4.057
8	-0.143	sel_in[3]	data_out_o[287]~reg0	clk_250	clk_250	4.000	-0.079	4.059
9	-0.131	sel_in[2]	data_out_o[315]~reg0	clk_250	clk_250	4.000	-0.090	4.036
10	-0.127	sel_in[2]	data_out_o[287]~reg0	clk_250	clk_250	4.000	-0.080	4.042
11	-0.108	sel_in[1]	data_out_o[42]~reg0	clk_250	clk_250	4.000	-0.088	4.015
12	-0.098	sel_in[0]	data_out_o[266]~reg0	clk_250	clk_250	4.000	-0.070	4.023
13	-0.071	sel_in[2]	data_out_o[289]~reg0	clk_250	clk_250	4.000	-0.075	3.991
14	-0.063	sel_in[0]	data_out_o[500]~reg0	clk_250	clk_250	4.000	-0.070	3.988
15	-0.057	sel_in[2]	data_out_o[179]~reg0	clk_250	clk_250	4.000	-0.100	3.952
16	-0.055	sel_in[3]	data_out_o[213]~reg0	clk_250	clk_250	4.000	-0.084	3.966
17	-0.050	sel_in[3]	data_out_o[399]~reg0	clk_250	clk_250	4.000	-0.088	3.957
18	-0.048	sel_in[1]	data_out_o[494]~reg0	clk_250	clk_250	4.000	-0.076	3.967
19	-0.048	sel_in[2]	data_out_o[61]~reg0	clk_250	clk_250	4.000	-0.083	3.960
20	-0.042	sel_in[1]	data_out_o[156]~reg0	clk_250	clk_250	4.000	-0.075	3.962
21	-0.037	sel_in[0]	data_out_o[436]~reg0	clk_250	clk_250	4.000	-0.082	3.950
22	-0.036	sel_in[0]	data_out_o[370]~reg0	clk_250	clk_250	4.000	-0.069	3.962
23	-0.032	sel_in[2]	data_out_o[421]~reg0	clk_250	clk_250	4.000	-0.085	3.942
24	-0.017	sel_in[3]	data_out_o[393]~reg0	clk_250	clk_250	4.000	-0.069	3.943
25	-0.010	sel_in[2]	data_out_o[481]~reg0	clk_250	clk_250	4.000	-0.068	3.937

Мультиплексор. Когда требования не сходятся

Setup



Путь блока от пина до регистра отправления.

Путь от выхода регистра отправления до регистра приема.

Путь блока от пина до регистра приема.

Tsetup

Насколько все плохо

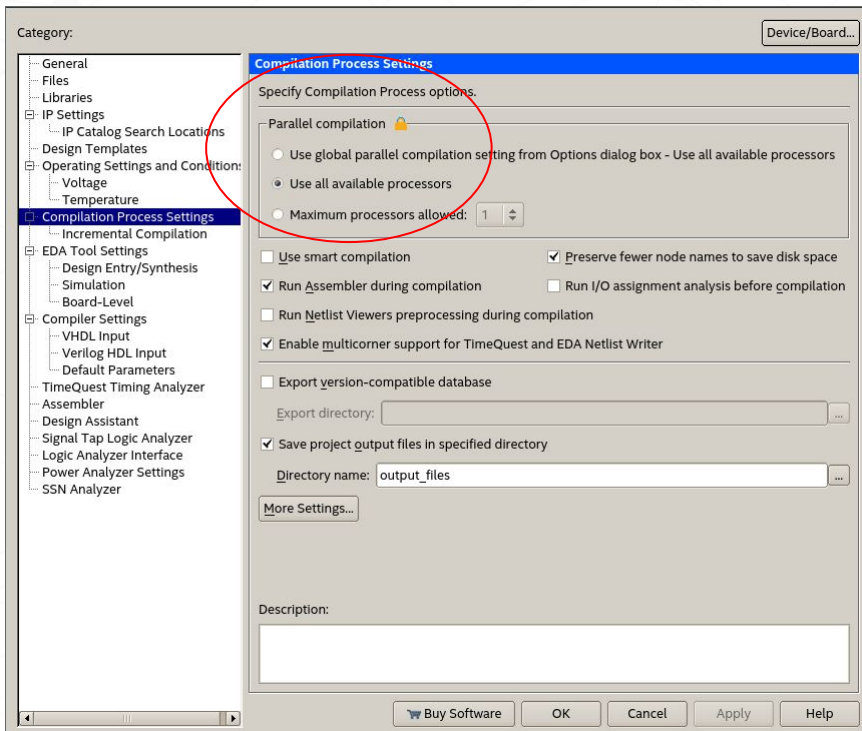
Мультиплексор. Когда требования не сходятся

Setup

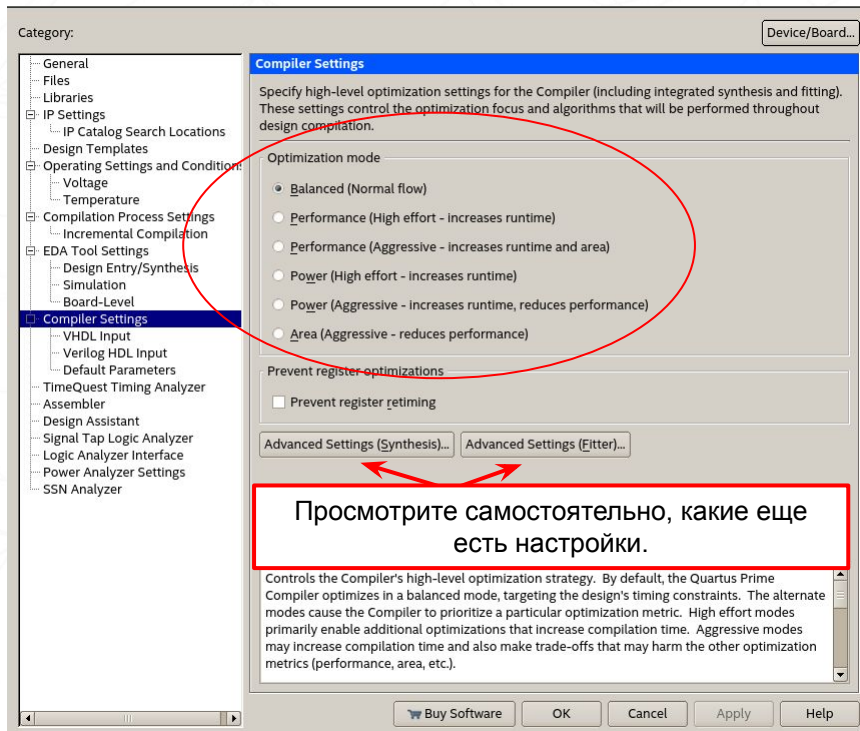
Data Arrival Path							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	▼ 2.203	2.203					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_R8	clk_i
3	0.000	0.000	RR	IC	1	IOIBUF_X27_Y0_N22	clk_i~input i
4	0.514	0.514	RR	CELL	1	IOIBUF_X27_Y0_N22	clk_i~input o
5	0.668	0.154	RR	IC	1	CLKCTRL_G18	clk_i~inputclkctrl inclk[0]
6	0.668	0.000	RR	CELL	8708	CLKCTRL_G18	clk_i~inputclkctrl outclk
7	1.684	1.016	RR	IC	1	FF_X28_Y16_N13	sel_in[2] clk
8	2.203	0.519	RR	CELL	1	FF_X28_Y16_N13	sel_in[2]
3	▼ 6.499	4.296					data path
1	2.402	0.199		uTco	1	FF_X28_Y16_N13	sel_in[2]
2	2.402	0.000	FF	CELL	1920	FF_X28_Y16_N13	sel_in[2] q
3	4.467	2.065	FF	IC	1	LCCOMB_X45_Y22_N20	mux_ins Mux198~0 datab
4	4.752	0.285	FF	CELL	1	LCCOMB_X45_Y22_N20	mux_ins Mux198~0 combout
5	4.990	0.238	FF	IC	1	LCCOMB_X45_Y22_N16	mux_ins Mux198~1 datab
6	5.359	0.369	FF	CELL	1	LCCOMB_X45_Y22_N16	mux_ins Mux198~1 combout
7	6.041	0.682	FF	IC	1	LCCOMB_X46_Y26_N24	mux_ins Mux198~9 dataa
8	6.408	0.367	FF	CELL	1	LCCOMB_X46_Y26_N24	mux_ins Mux198~9 combout
9	6.408	0.000	FF	IC	1	FF_X46_Y26_N25	data_out_o[313]~reg0 d
10	6.499	0.091	FF	CELL	1	FF_X46_Y26_N25	data_out_o[313]~reg0

Высокий fanout (количество потребителей сигнала) обычно приводит к проблемам из-за удаленности регистров друг от друга.

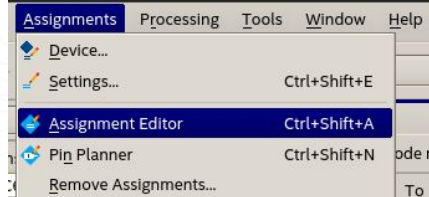
V. Настройки проекта Quartus, назначение пинов



Увеличения количества используемых процессоров не сильно ускорит сборку проекта.



Режим оптимизации: максимальная частота, минимальное потребление, минимальное количество ресурсов.



“Виртуальный” пин -- сигналы, действующие на пин не оптимизируются, но привязки к реальному пину нет. Полезно при сборке отдельных модулей (для оценки ресурсов и частоты).

Pin	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓	in sel_i	Virtual Pin	On	Yes	mux_top		
2	✓	data_0_i	Virtual Pin	On	Yes	mux_top		
3	✓	data_1_i	Virtual Pin	On	Yes	mux_top		
4	✓	data_out_o	Virtual Pin	On	Yes	mux_top		
5	✓	in clk_i	Location	PIN_R8	Yes			
6		<<new>>	<<new>>					

Pin	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓	in sel_i	Virtual Pin	On	Yes	mux_top		
2	✓	data_0_i	Virtual Pin	On	Yes	mux_top		
3	✓	data_1_i	Virtual Pin	On	Yes	mux_top		
4	✓	data_out_o	Virtual Pin	On	Yes	mux_top		
5	✓	in clk_i	Location	PIN_R8	Yes			
6		<<new>>	<<new>>					

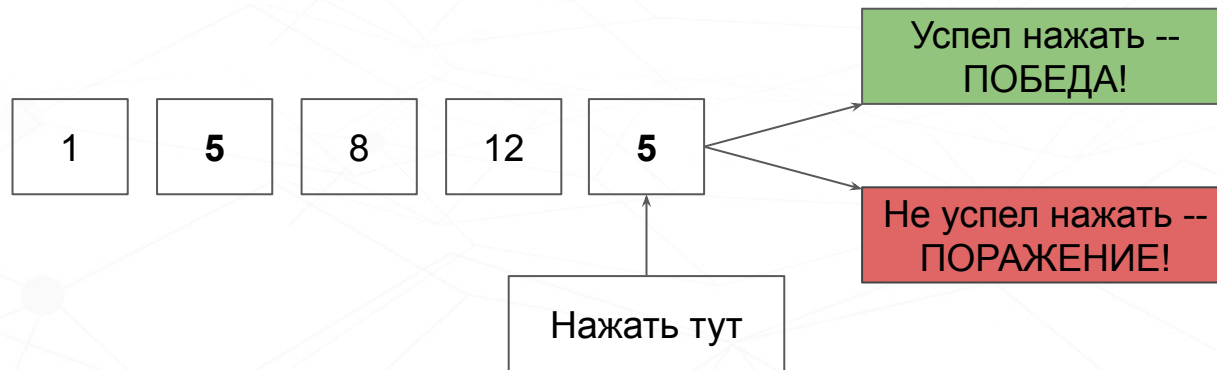
Двойной клик -- пишем имя сигнала.

Двойной клик -- и не пугаемся (список огромный). Основные варианты, которые нужны:

- Virtual Pin -- объявляем сигнал виртуальным.
- Location -- задаем пин. В Value указываем имя пина. Тактовый сигнал лучше всегда назначать на реальный пин.

VI. Собираем проект игры “N-back”

Нужно нажать на кнопку, когда показывается число, которое было 3 символа назад.



Отладочная плата DE0-Nano

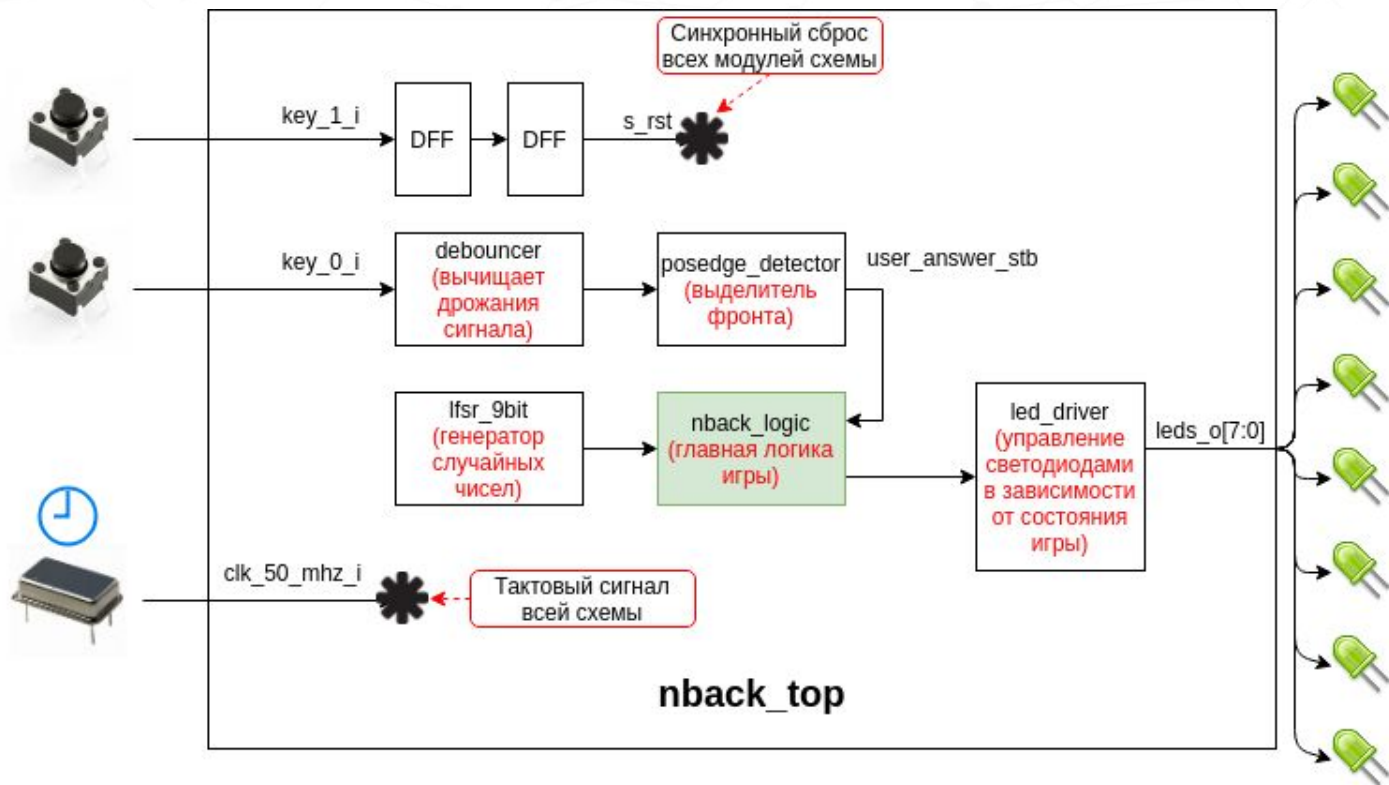


- Altera Cyclone IV EP4CE22F17C6N FPGA
- 8 x LEDs
- 2 кнопки
- 4 переключателя
- G-Sensor ADI ADXL345
- Clock 50 МГц

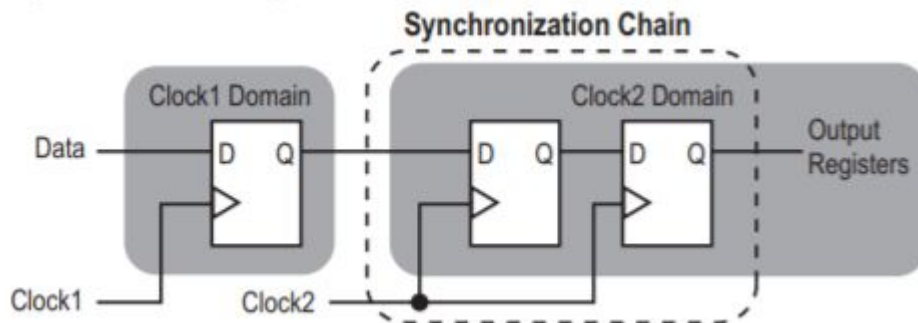
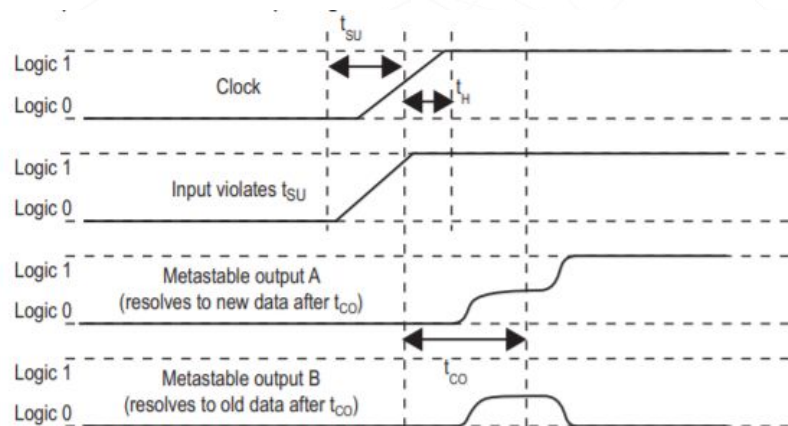


Решение проблемы с драйвером USB Blaster на Windows 10: [Altera USB Blaster Driver Installation Instructions](#)

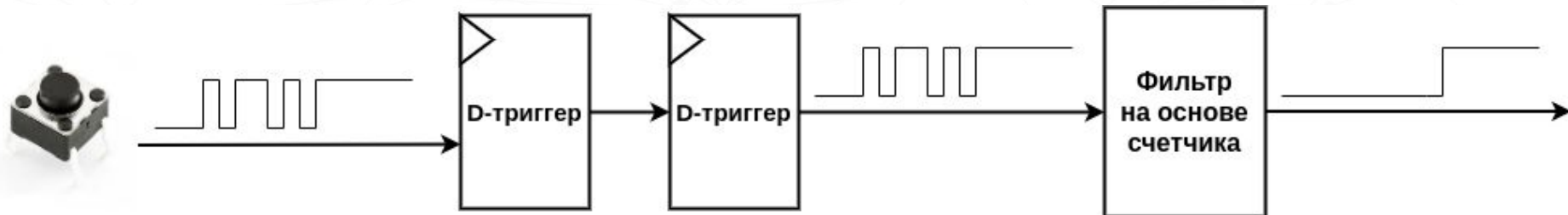
<https://github.com/stcmtk/fpga-webinar-2020/tree/master/project>



Understanding Metastability in FPGAs



Проект на github: https://github.com/stcmtk/fpga-webinar-2020/tree/master/lecture_3/debouncer



Не синхронный с тактовой частотой и **с помехами** сигнал.

Синхронизация с тактовой частотой.
2 триггера:
Первый может попасть в **метастабильное** состояние (не 0 и не 1) и это состояние может попасть дальше в схему.

Синхронный с тактовой частотой, но все еще с помехами.

Сигнал должен не изменяться в течении заданного времени -- тогда считаем что значение стабильное.

Синхронный с тактовой частотой и чистый сигнал.

// Полином: $x^9 + x^5 + 1$

parameter WIDTH = 9;

parameter TAP1 = WIDTH-1;

parameter TAP2 = 4;

logic [WIDTH-1:0] prbs_state;

always_ff @(posedge clk_i)

if(s_rst_i)

prbs_state <= (WIDTH)'(1);

else

begin

if(ack_i)

begin

prbs_state[0] <= prbs_state[TAP1] ^ prbs_state[TAP2];

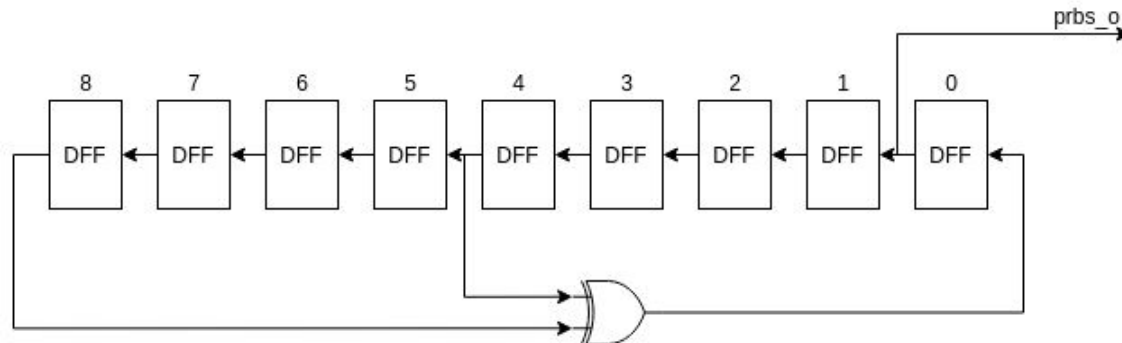
prbs_state[WIDTH-1:1] <= prbs_state[WIDTH-2:0];

end

end

assign prbs_o = prbs_state[0];

До сброса prbs_state = 0
Последовательности не
будет.
После прошивки нужен
сброс.

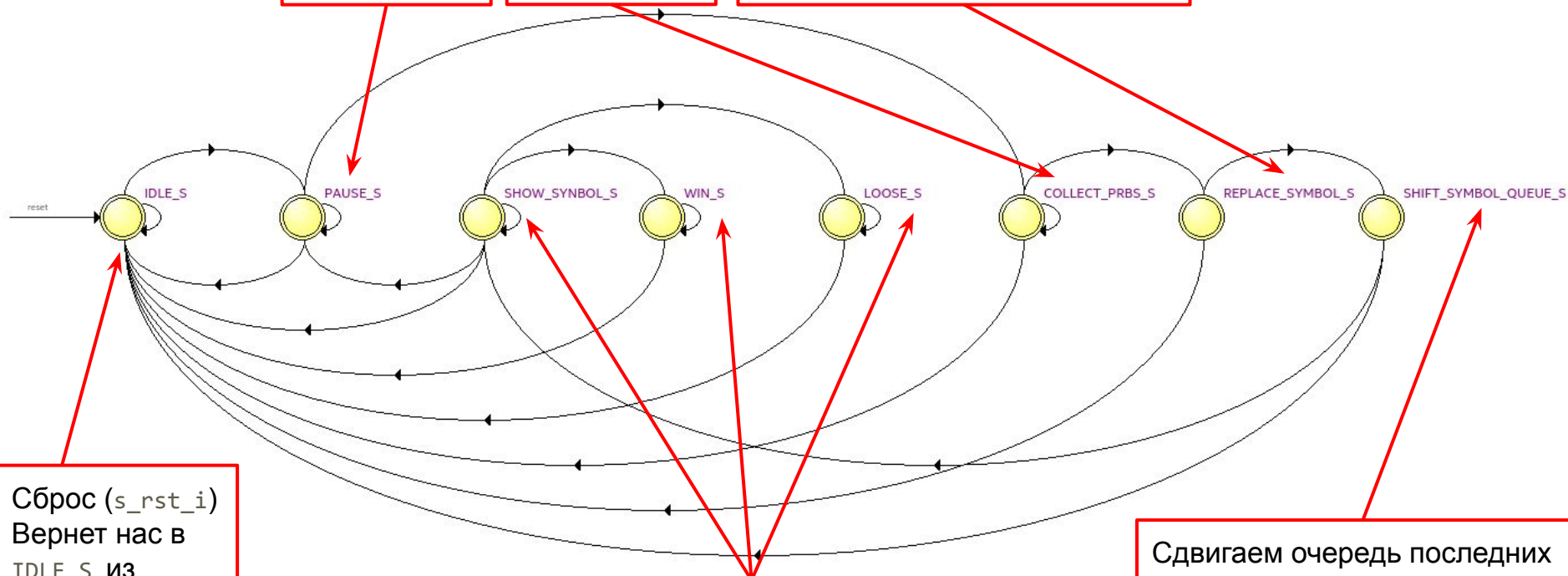


Пауза между
показом
символов

Собираем 8
бит символа
по 1 биту от
LFSR

Чтобы игра была интереснее:
с вероятностью 25%
подменяем символ на тот,
который был N назад.

Логика игры



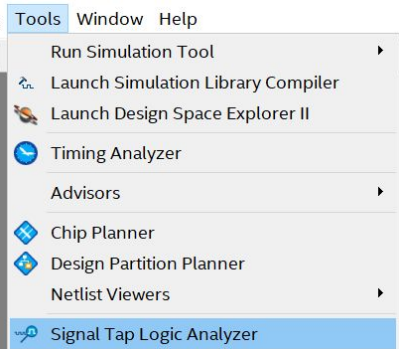
Сброс (`s_rst_i`)
Вернет нас в
`IDLE_S` из
любого
состояния

Показываем символ и проверяем, нажал ли
пользователь кнопку. Если да, принимаем
решение -- выиграл или проиграл.

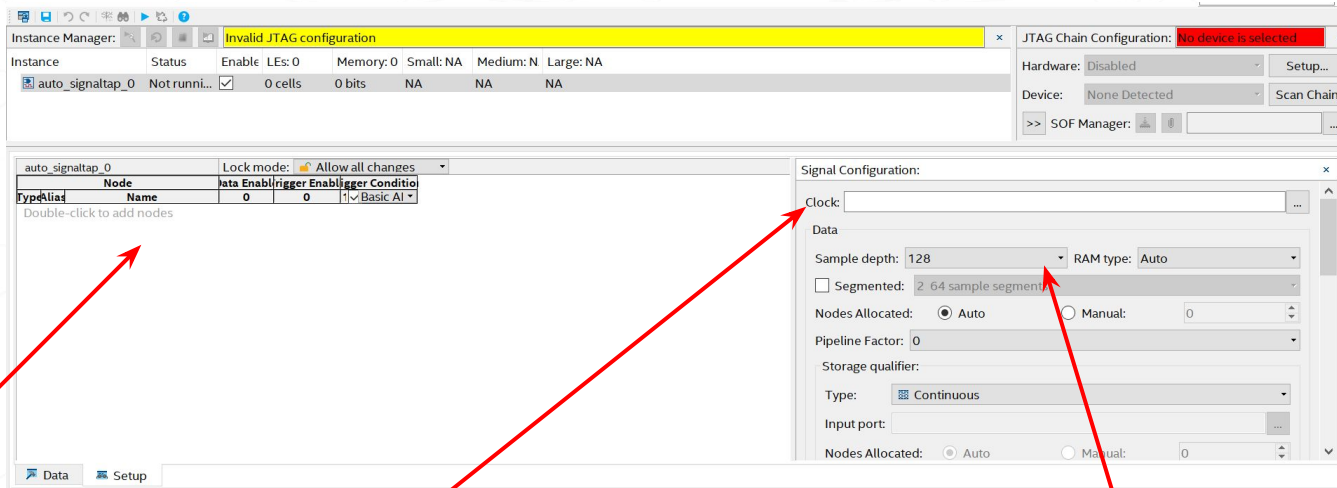
Сдвигаем очередь последних
N+1 символов.

Параметр	Комментарий	Значение в проекте
N	n в “n-back”	3
SYMBOL_W	Разрядность одного символа.	8
SYMBOL_DURATION_HW_TICKS	Длительность показа символа в тактах блока	200000000 (4 секунды при частоте 50 МГц)
PAUSE_DURATION_HW_TICKS	Длительность паузы между символами в тактах блока	200000000 (4 секунды при частоте 50 МГц)

Отладчик позволяет смотреть на сигналы внутри работающей схемы.



Добавляем сигналы, на которые хотим смотреть. Чем больше сигналов, тем больше будет занято ресурсов.



Задаем тактовый сигнал. Все сигналы, которые мы добавили слева должны быть синхронны с этим сигналом.

Задаем размер выборки. Сколько тактов подряд сможем смотреть.